PCT

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04J 3/24, H04L 29/06, H04N 7/58	A1	(11) International Publication Number: **WO 99/56423** (43) International Publication Date: 4 November 1999 (04.11.99)

(21) International Application Number: PCT/GB99/01218

(22) International Filing Date: 21 April 1999 (21.04.99)

(30) Priority Data:
9809006.1   29 April 1998 (29.04.98)   GB

(71) Applicant *(for all designated States except US)*: UNIVERSITY OF STRATHCLYDE [GB/GB]; McCance Building, 16 Richmond Street, Glasgow G1 1XQ (GB).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: LAMBERT, Robert, Bartholomew [GB/GB]; 219 Ritchie Park, Johnstone, Renfrewshire PA5 8JL (GB). FRYER, Richard, John [GB/GB]; 2 Kings Park, Torrance, Glasgow G64 4DX (GB). BRESLIN, David, James [GB/GB]; 21 Huntly Drive, Cambuslang, Glasgow G72 8PU (GB). DEVLIN, Paul, Edward, George [GB/GB]; 7 Dick Street, Kelvinbridge, Glasgow G20 6JF (GB).

(74) Agents: McCALLUM, William, Potter et al.; Cruikshank & Fairweather, 19 Royal Exchange Square, Glasgow G1 3AE (GB).

(81) Designated States: CA, CN, GB, JP, KR, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: MULTIPLEXER/DE-MULTIPLEXER
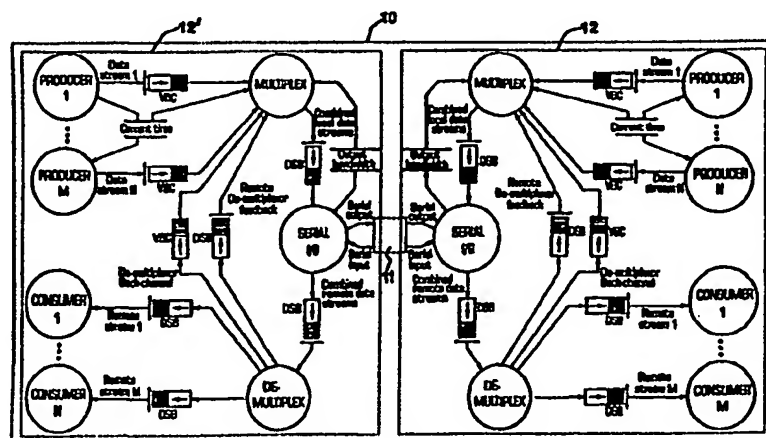
(57) Abstract

A multiplexer for transmitting real–time data from a plurality of producers to a plurality of corresponding consumers over a channel, where each producer is operable to present a data block for transmission, so that at any point in time a plurality of data blocks are presented for transmission. The multiplexer is arranged in use to create and transmit across the channel a data stream comprising a sequence of variable length frames, where each frame contains either all or part of one of the plurality of data blocks and does not contain data from any other data block. The variable frame length enables the multiplexer to transmit entire data blocks a greater percentage of the time than if the multiplexer was constrained to a predetermined frame length. This reduces fragmentation of data blocks. The multiplexer can change the frame length in response to any changes in the channel bandwidth. The multiplexer is able to transmit small frames when the system is operating near to its bandwidth limit which may occur when the channel bandwidth drops temporarily because of burst errors. The variable length frame is essentially only as long as the data block it conveys.

1

MULTIPLEXER/DE-MULTIPLEXER

The present invention relates to a multiplexer for transmitting real-time data, and to a de-multiplexer for receiving the transmitted data. The present invention also
5   relates to a data transmission system incorporating such a multiplexer and de-multiplexer.

Multiplexer/de-multiplexers are used to route data from several producers through a single channel to respective consumers so that each producer sends data
10   blocks to its corresponding consumer. The blocks of data from the different producers are integrated (multiplexed) to form a single data stream which is conveyed by the channel. Once the data stream has been conveyed, the blocks of data are extracted (de-multiplexed) from this
15   single stream and each block of data is applied to the corresponding consumer. This integration and extraction process must be invisible to the producers and consumers.

When transmitting real-time data which are in variable-sized data blocks, such as audio and video data,
20   it is conventional to integrate the data blocks from the different producers into fixed size frames, and then to transmit the fixed size frames across a high bandwidth channel so that the data can be transferred with minimal delay. Each fixed size frame contains predetermined
25   amounts of data from the data blocks sent by each producer, for example, a fixed size frame may contain 40 bits from a first producer, 25 bits from a second producer, and 20 bits from a third producer.

However, delays are introduced with this system. For
30   example, when the data block from the first producer has less than 40 bits, the remaining bits (to make up the 40 bits) are filled with meaningless filler digits which consume assembly time and transmission time. Also if the data block from the first producer is 45 bits, then 5 bits
35   are carried over into a second frame and the other 35 bits are either filled with meaningless filler digits (which wastes bandwidth), or the assembly of the fixed length

2

frame is delayed until the first producer sends another 35
bits (which introduces a delay).

To transmit real-time data efficiently the
transmission must have a low overhead, that is, a low
percentage of the channel must be used for routing and
control information; and low latency, that is, low delay
between providing data for transmission and the data being
received by the intended recipient. All presently known
multiplexer/de-multiplexer systems for real-time data
transmission have a high overhead and/or a high latency.

It is an object of an embodiment of the present
invention to provide an improved multiplexer and a data
transmission system incorporating said improved
multiplexer.

According to a first aspect of the present invention
there is provided a multiplexer for transmitting real-time
data from a plurality of producers to a plurality of
corresponding consumers over a channel, where each producer
is operable to present a data block for transmission, so
that at any point in time a plurality of data blocks are
presented for transmission,

the multiplexer being arranged in use to create and
transmit across the channel a data stream comprising a
sequence of variable length frames, where each frame
contains either all or part of one of the plurality of data
blocks and does not contain data from any other data block.

By virtue of the variable frame length the multiplexer
is able to transmit entire data blocks a greater percentage
of the time than if the multiplexer was constrained to a
predetermined frame length. This reduces fragmentation of
data blocks which is a cause of high transmission
overheads. In addition, by using a variable frame length
the multiplexer can change the frame length in response to
any changes in the channel bandwidth. Thus, the
multiplexer is able to transmit small frames when the
system is operating near to its bandwidth limit which may
occur when the channel bandwidth drops temporarily because

3

of burst errors. Another advantage of having a variable length frame is that the frame is essentially only as long as the data block it conveys.

It will be appreciated that not all of the plurality of producers may be active, that is, not all of the plurality of producers may present data for transmission at every point in time.

Preferably, a variable length frame (VLF) comprises a predetermined code, an identifier for indicating which consumer the data is intended for, and the data to be transmitted.

The predetermined code may be in the form of a producer code which is used to indicate that data from a producer is being transmitted, or the predetermined code may be in the form of a control code which is used to indicate that control data is being transmitted. Conveniently the control code is equal to two successive occurrences of the producer code.

Preferably, each variable length frame is byte aligned so that the predetermined code begins at the start (the first bit) of a byte.

Preferably, bit stuffing and/or byte replacement are used to avoid accidental occurrence of the producer code.

Preferably, the multiplexer is incorporated in a multiplexer/de-multiplexer data transmission system for duplex transmission and reception of data.

Preferably, the data transmission system further comprises buffer means associated with each of the plurality of producers, where the buffer means stores data for transmission.

Preferably, at least one producer is an adaptive producer which is operable to regulate either the time interval at which it presents data blocks to the buffer means for transmission, and/or the size of data block which it presents to the buffer means for transmission. More preferably, substantially all of the producers are adaptive producers.

4

Preferably, the adaptive producer is operable to interact with the buffer means and thereby to determine the effective bandwidth of the buffer means (that is, the rate at which data is transferred from the buffer means to the multiplexer), whereby, in use, prior to writing a data block to the buffer means the adaptive producer interrogates the buffer means to determine the effective bandwidth of the buffer means, and the adaptive producer accordingly controls the size of data block to be written to the buffer means and/or the rate at which data blocks are written.

It will be appreciated that if the buffer means has a high effective bandwidth the adaptive producer increases the data block size and/or the rate at which data is written; whereas, if the buffer means has a low effective bandwidth the adaptive producer decreases the data block size and/or the rate at which data is written.

Preferably, each data block written to the buffer means has a data identifier. Conveniently, the data identifier indicates the time the block was written, and the size of the block.

The advantage of having a data identifier is that the multiplexer can scan the data identifiers associated with each data block presented by the buffer means and thereby determine for each data block how long the block has been awaiting transmission and how many bytes would have to be transmitted to transmit the entire data block.

Preferably, the buffer means comprises: a buffer for holding data blocks which are to be transmitted, and a memory for holding the data identifiers associated with the data blocks in that buffer; the buffer means having a FIFO (first in first out) arrangement so that each buffer means presents only one data block and the associated identifier at a time (the current block and identifier) to the multiplexer.

Conveniently, the buffers are capable of receiving blocking writes.

5

Preferably, each of the buffer means has an associated buffer identifier. Conveniently, the buffer identifier indicates the relative priority and the type of data conveyed by that buffer.

Preferably, the data transmission system is operable to change the buffer identifier so that, in use, the relative priority of each buffer may be modified, for example, in response to changes in the channel bandwidth.

Preferably, the multiplexer is adaptive and communicates with the transmission channel via a variable bandwidth FIFO buffer and is operable to regulate the time interval at which it presents VLFs to the buffer and/or the size of VLFs according to the effective bandwidth of the variable bandwidth FIFO buffer.

Preferably, the multiplexer reads the data identifier and the buffer identifier associated with each current block to determine which current block or part of which current block should be formed into the next variable length frame. Conveniently, the multiplexer implements a predetermined strategy for determining the order in which the data blocks should be formed into VLFs for transmission. The predetermined strategy preferably is a function of channel bandwidth (which may vary from time to time), buffer identifier (which may be preset or vary from time to time), and data block identifier (which indicates the block length which may vary from buffer to buffer and also may vary from time to time in any one buffer).

The advantage of implementing a predetermined strategy is that time-critical data, for example video and corresponding audio, can be transmitted quasi-simultaneously (very little time difference between transmitting the video and the audio) at the expense of data that is not time-critical.

According to another aspect of the present invention there is provided a method of transmitting real-time data from a plurality of producers over a channel, the method comprising the steps of:

6

conveying from each of the producers a data block to be transmitted from one of the producers;

creating a variable length frame using one conveyed data block; and

transmitting the variable length data frame across the channel.

Preferably, the creating step further comprises the step of interrogating the identifier associated with each presented data block to determine the data block which has highest priority according to a predetermined strategy and selecting this highest priority data block for conveying.

These and other aspects of the present invention will be apparent from the following description, given by way of example, with reference to the accompanying drawings in which:

Fig 1 is a schematic diagram of a communications system in accordance with one embodiment of the present invention;

Fig 2 is a schematic diagram of a data transmission system which is a part of the communications system of Fig 1;

Fig 3 is a schematic diagram of the multiplexer component of Fig 2;

Fig 3A is a flowchart illustrating one example of a predetermined strategy implemented by the multiplexer component of Fig 2;

Fig 4 is a diagram which illustrates the implementation of the buffer component of Fig 2;

Fig 5 is a schematic diagram of the format of a variable length data frame used in the embodiment of Fig 2;

Fig 6 is a schematic diagram of the format of a variable length control frame used in the embodiment of Fig 2;

Figs 7a and 7b are tables illustrating details of the variable length control frame of Fig 7;

Figs 8a and 8b illustrate the byte replacement and the bit stuffing arrangements used in the embodiment of Fig 2;

Fig 9 illustrates the creation of a variable length data frame such as would be used in the embodiment of Fig 2;

Fig 10 illustrates the method of avoiding conflicts in creating a variable length data frame, using direct bit replacement to prevent accidental occurrences of a predetermined code;

Fig 11 illustrates the closing sequence of a variable length data frame;

Fig 12 is a schematic diagram of the de-multiplexer component of Fig 2; and

Fig 13 is a schematic diagram of an alternative data transmission system.

Fig 1 is a schematic diagram of a communications system 10 intended for duplex transmission of real-time data over a channel 11 which interconnects two similar data transmission and reception systems 12,12'.

Referring to Fig 2, which shows one of the data transmission and reception systems 12 in more detail, there is a multiplexer 13 (for data transmission) and a de-multiplexer 14 (for data reception). The multiplexer 13 is connected to a plurality of producers 16a to 16n (only two of which are shown) via a corresponding plurality of buffer means 18 in the form of variable bandwidth channel (VBC) buffers 18a to 18n.

The de-multiplexer 14 is connected to a plurality of consumers 20a to 20m via a corresponding plurality of buffer means 22a to 22m in the form of data stream buffers (DSBs) which are serial buffers having a FIFO arrangement.

The multiplexer 13 and the de-multiplexer 14 are connected via a serial input/output channel (serial I/O) 24, which is connected to data transmission system 12' via channel 11. Data transmission system 12 is hereinafter referred to as the local system and data transmission system 12' is hereinafter referred to as the remote system. Therefore, when a component in the remote system 12' is designated the numeral of the corresponding component in

the local system 12 will be used but with the addition of the prime symbol ', thus producer 16a is in local system 12 but producer 16a' is in remote system 12'. In use, the producers 16 of local system 12 transmit data to the consumers 20' of remote system 12'; and the producers 16' of remote system 12' transmit data to the consumers 20 of local system 12.

In one embodiment, the channel 11 is, for example, a GSM cellular telephone line having a nominal bandwidth of 7600 bits per second. However, serial I/O 24 transmits and receives data at the actual (or instantaneous) bandwidth of the channel 11.

Each VBC buffer 18 emulates a variable bandwidth communications link. The rate at which data is removed from the VBC 18 determines the effective bandwidth of the buffer. As will be described in more detail below, each producer 16 is adaptive, which means that each producer (e.g. 16a) can interact with its corresponding VBC 18a to obtain information on the current bandwidth of the VBC 18a, so that the producer 16a can self-regulate either the time interval at which it presents data blocks to the VBC 18a and/or the size of data blocks.

The producers 16 may be video image compressors, still image compressors, audio compressors, status monitor devices, actuator feedback monitor devices, or some other type of data generator or compressor. Different producers 16 may generate data having different characteristics. For example, the data may be time-critical fixed bandwidth data if a fixed rate speech codec is used as a producer; time-critical non-controlled variable bandwidth if a speech codec with silence detection is used as a producer; time-critical controlled variable bandwidth if a video codec is used as a producer; or non time-critical data if a text file is being conveyed by a producer.

In use, each producer 16 (for example 16a) receives data from a respective source 30, compresses or otherwise arranges the data, and presents a data block (of a size

9

determined by producer 16a) to the corresponding VBC 18a
for multiplexing to integrate with data blocks from some or
all of the other producers 16b to 16n. In the example that
follows, it will be assumed that all n producers 16 are
5 active and continually convey data blocks to the respective
VBCs 18 for transmission.

Each data block that is presented to a VBC 18 has an
associated data identifier which indicates characteristics
of the data in the data block. In this embodiment, the
10 data identifier indicates the size of the data block and
the time that the data block was written to the VBC.

Each VBC 18 has a FIFO arrangement so that each VBC 18
presents only one data block to the multiplexer 13 at a
time. Each VBC 18 also has an associated (fixed) buffer
15 priority which indicates the relative priority of the VBC
18 compared with the other VBCs 18.

The multiplexer 13, in its basic operation, scans the
data identifier associated with each data block presented
by each VBC 18 and determines in accordance with a
20 predetermined strategy which of the n data blocks presented
has the highest priority. The multiplexer 13 then reads
the data block which has the highest priority and creates
a variable length frame (VLF) which incorporates that data
block. The length of the frame being selected so that it
25 is only sufficiently long to accommodate the particular
data block and any codes which require to be transmitted as
part of the frame (as will be further explained below).
The multiplexer 13 then sends the VLF for transmission
across the channel 11.

30 The multiplexer 13 repeatedly performs the above steps
of scanning all of the data identifiers; determining which
data block has the highest priority; creating a VLF
incorporating the data block; sending the VLF for
transmission across the channel 11; so that a continuous
35 data stream comprising a sequence of VLFs is created.

After each data block is read by the multiplexer 13,
the VBC 18 advances the next data block and the associated

data identifier so that the next data block and associated identifier are automatically presented to the next scan of the multiplexer 13 by the VBC 18.

Each data frame generally corresponds to one data block; however, in some circumstances (as will be explained) the data block may be split so that only a part of the data block is included in one VLF and the remainder of the block is included in a different VLF. This may occur, for example, when a large low priority data block (such as a text file) is being transmitted. Another example of a situation in which a data block may be split is when two equally high priority large data blocks are to be transmitted; in such a case the multiplexer 13 may transmit a small portion from each data block alternately. Another example of such a situation is when the channel bandwidth drops dramatically and the multiplexer 13 accordingly reduces the size of VLF which it conveys; in which case data blocks may be split.

In the case where only part of a data block is incorporated into a VLF the multiplexer 13 effectively updates the data identifier associated with the remaining data block portion.

Each VLF is formed from data from only one VBC 18. The data stream is continuous because the multiplexer 13 is continually creating VLFs using the data blocks presented by the VBCs 18. The data stream will only stop if all of the VBCs 18 are empty.

The sequence of VLFs is conveyed to the serial I/O 24 via a buffer 32. Buffer 32 is only capable of receiving blocking writes (indicated in Fig 2 by an arrowhead and a bar immediately in front of the arrowhead). When a blocking write is requested, if the buffer 32 is full then the write will wait until there is sufficient space in buffer 32 for the write operation to be performed.

As the buffer 32 fills up the multiplexer 13 responds immediately and self-regulates by reducing either the size and/or the frequency of the VLFs it conveys to the buffer

32. This, in turn, causes a reduction in the effective bandwidths of the VBCs 18, which causes the producers 16 to reduce the data block size they present to the VBCs 18 and/or the rate at which they present data blocks to the VBCs 18. During this self-regulating action of the multiplexer 13 the data block from the VBC 18 may be split.

The serial I/O 24 transmits the sequence of VLFs to a corresponding serial I/O in system 12' (serial I/O 24') via the channel 11. Serial I/O 24 may have a channel bandwidth detector for determining both the nominal and the actual (i.e. instantaneous) bandwidth of channel 11 as the channel 11 is operating, and for feeding this information back to the multiplexer 13 along line 70 to set the initial operation of the multiplexer 13 (the rate and size of VLFs conveyed to the DSB 32) and to anticipate the bandwidth measure provided dynamically to the multiplexer 13 by the DSB 32.

In system 12', serial I/O 24' receives the sequence of VLFs and conveys this sequence to the de-multiplexer 14' via a buffer 34'. The de-multiplexer 14' retrieves the data blocks from the sequence of VLFs and writes each data block to the respective DSBs 22'. Each consumer 20' then reads the data block from the corresponding DSB 22'.

System 12 has a backchannel 40 to allow the de-multiplexer 14 to communicate control information to the remote multiplexer 13', for example to acknowledge safe receipt of data, or to convey commands or status information (for example, local consumer 20a is full). The backchannel 40 is in the form of a VBC 42 directly connected between the multiplexer 13 and the de-multiplexer 14 so that the de-multiplexer 14 can present control information as a data block to the multiplexer 13 for incorporating into a VLF.

System 12 also has a feedback channel 44 for conveying feedback information from the remote de-multiplexer 14' to the local multiplexer 13 for the purpose of modifying the function of multiplexer 13.

12

Thus, channel 40 data travels via multiplexer 13, de-multiplexer 14', and channel 44' to multiplexer 13', whilst channel 40' data travels via multiplexer 13', de-multiplexer 14 and channel 44 to multiplexer 13.

5   If the de-multiplexer 14' attempts to write a data block to one of the consumer DSBs (for example 22b') and the write fails because consumer DSB 22b' is full, then all future data blocks which are intended for DSB 22b' are discarded and the de-multiplexer 14' sends a command via
10   the backchannel 40 to instruct the multiplexer 13 to cease reading data blocks from local producer 16b. Producer 16b, in consequence, stops presenting data to VBC 18b because the effective buffer bandwidth has dropped to zero.

Fig 3 is a more detailed schematic diagram of the
15   multiplexer 13 of the system 12 of Fig 2. In Fig 3, there are two principal components, a channel arbiter 60 and an integrator 64.

The multiplexer 13 must integrate data blocks from multiple VBCs 18 into a combined stream so that the
20   combined stream can be transmitted over the channel 11. The multiplexer 13 must keep real-time interactive data blocks in close temporal relation so that there is minimal delay between the interactive data blocks. Interactive data is data which combines with other data, such as video
25   and corresponding audio, where the audio must be synchronised with the corresponding video so that the movements of a person's lips is synchronised with their speech. The multiplexer 13 must also ensure that time-critical data is integrated and transmitted as quickly
30   as the channel 11 permits. The multiplexer 13 also attempts to transfer entire data blocks in a single transfer because fragmentation of data blocks causes a high transmission overhead. Additionally, the bandwidth of channel 11 is not fixed and may vary very significantly
35   from its nominal value.

Accordingly, the channel arbiter 60 has the responsibility of implementing a predetermined strategy for

determining the order in which the data blocks should be formed into VLFs for transmission. It does this by assigning actual bandwidth to each of the active producers 16 that are currently transmitting data based on the data identifier (statistics) 62 of each active VBC 18 and a buffer identifier. In this embodiment each buffer identifier has a (fixed) predetermined priority based on the type of data generated by its associated producer.

As regards bandwidth assignment, the channel arbiter may determine that one active producer 16 will be given at least a minimum bandwidth (for example a producer 16a generating audio data may be given at least 2000 bits per second); whereas, another active producer 16b may be given another minimum bandwidth (for example, at least 1000 bits per second). This has the advantage that even if the channel bandwidth drops substantially from its nominal value the most important data can still be conveyed and the less important data does not overload or congest the transmission system.

The channel arbiter 60 has a time input 68 conveying the current time so that the current time can be compared with the time at which each data block in each VBC 18 was written to determine how long each data block has awaited transmission.

The arbiter 60 also has an output bandwidth input 70 which is received from the serial I/O 24. The output bandwidth input 70 conveys information on the actual current bandwidth which is used by the serial I/O 24. The actual current bandwidth of channel 11 may differ from the nominal bandwidth of channel 11 because of burst errors. If the actual current bandwidth of channel 11 drops then the multiplexer 13 reduces the size and/or rate of the VLFs as described above.

The arbiter 60 also receives a feedback input 72 indicating the number of bytes written to the DSB 32. The arbiter 60 has a selection output 80 by which the arbiter 60 conveys information about which of the VBCs 18 (or 42)

should be selected for transmitting data and how much data should be transmitted from the selected VBC 18.

The integrator 64 selects the data block or partial data block from the selected VBC 18 (or 42) in response to the information on the selection output 80. The integrator 64 then forms a VLF from the selected data block and outputs the VLF on the combined sequence output 90. The sequence of VLFs is then conveyed to the serial I/O 24 via buffer 32 as described above.

Fig 3A is a flowchart illustrating one example of a predetermined strategy for determining the order in which data blocks should be formed into VLFs for transmission. This strategy uses flow steps 65a through 65z described in Fig 3A and which will now be explained.

When the multiplexer 13 is to send the first block of data an initiation procedure (steps 65a,b) is implemented. The arbiter 60 assigns to each of the n VBCs 18 an individual bit budget (step 65b) which is a value equivalent to a number of bits. The values of the individual bit budgets assigned are not critical, but will generally depend on the nominal bandwidth of the system and the maximum size of a VLF (that is, the maximum number of bits in a data block which the multiplexer 13 will allow to be formed into a single VLF). The values of the individual bit budgets, though not critical, should be selected so that very small values and very large values are avoided, as will be explained later.

The arbiter 60 also determines for each VBC 18 the minimum bandwidth required for the data in that VBC 18 (step 65b) based on the data type information (video, text, and such like) stored in the VBC's 18 buffer identifier.

The channel arbiter 60 then determines the instantaneous channel bandwidth (step 65c). If no data block has been transmitted previously then the instantaneous channel bandwidth may be set to the nominal channel bandwidth or some fraction thereof. If a data block has been transmitted previously then the

15

instantaneous channel bandwidth is obtained from output bandwidth line 70 or from the time taken to transmit the last VLF.

In response to the determined instantaneous channel bandwidth value, the multiplexer 13 may update the maximum VLF value (step 65d). For example, if the channel bandwidth has dropped dramatically then the maximum VLF value may be reduced.

The n different VBCs 18 are then ordered (step 65e) using the n buffer identifiers so that the data types are prioritised. For example, a VBC 18 having video data may be ordered before (given a higher priority than) a VBC 18 having audio data which may be ordered before a VBC 18 having text data.

For those VBCs 18 having data for transmission, beginning with the highest priority VBC 18 (as determined by step 65e) and successively working towards the lowest priority VBC 18, the arbiter 60 selects (steps 65f, 65g) each VBC 18 having a minimum bandwidth less than the current instantaneous bandwidth whilst sequentially subtracting the value of the minimum bandwidth of the VBC 18 selected from the current instantaneous bandwidth to produce a set of selected VBCs (step 65h). As each selected VBC 18 is added to the set of selected VBCs 18 a new (smaller) current instantaneous bandwidth is generated.

The current instantaneous bandwidth is indicated in Fig 3A by the variable AVAIL. After the ordering stage (step 65e) the variable AVAIL is set to equal the instantaneous bandwidth (step 65f). After each VBC 18 is selected, the minimum bandwidth of that VBC 18 is subtracted (step 65i) from the current value of AVAIL. It is the current value of AVAIL that is termed the current instantaneous bandwidth.

The arbiter 60 ignores (does not select) any VBC 18 having a minimum bandwidth greater than the current instantaneous bandwidth for that VBC 18. If the current instantaneous bandwidth is reduced to zero then the arbiter

16

60 ignores all VBCs 18 having lower priority than the VBC 18 whose selection caused the current instantaneous bandwidth to be reduced to zero.

Steps 65k and 65l are used to ensure that the VBCs 18 are operated on (i.e. selected or ignored) in the order of their priority as determined by step 65e.

For each of the selected VBCs 18, the arbiter 60 determines (step 65m) the relative size of the minimum bandwidth for that VBC 18 to the sum total of the minimum bandwidths of all of the selected VBCs 18 so that a division percentage (or apportionment) can be generated for each selected VBC 18.

The arbiter 60 then determines whether a VLF has been written previously (step 65n). If not then the arbiter 60 jumps to step 65p which is the beginning of the sequence which determines which data block should be formed into a VLF. If, however, a VLF has been written previously, then the arbiter distributes (apportions) (step 65o) the number of data bits used in the data block of the previous VLF to each of the selected VBCs 18 based on the division percentage (step 65m) of each selected VBC 18.

For each selected VBC 18, the apportioned data bits are added to the individual bit budgets so that the individual bit budgets are augmented. For example, if the bit budget of VBC 18a is currently 750 bits, and 500 bits were transmitted (from VBC 18b) in the last VLF, and VBC 18a has a 50% apportionment, then the bit budget of VBC 18a will be increased to 1000 bits.

The arbiter 60 then begins the sequence which determines which data block should be formed into a VLF. This sequence involves steps 65p through 65v.

For each of the selected VBCs 18 the arbiter 60 evaluates whether the data block to be transmitted is larger than the maximum VLF value (steps 65q,r). If the data block is larger than the maximum VLF value then the maximum VLF value is used to represent the size of data portion to be transmitted (indicated in step 65r by

17

variable 'P'); whereas, if the data block is smaller than the maximum VLF value then the data block size is used as variable 'P'.

5 The arbiter 60 then determines whether the 'P' variable value (which is the lower of the maximum VLF value and the actual data block size) for each selected VBC 18 is lower than the individual bit budget value for that selected VBC 18 (step 65s).

The arbiter 60 then chooses from the determined VBCs
10 18 having a 'P' value lower than the individual bit budgets, the VBC 18 having the highest wait time (steps 65s,t). The wait time is the difference between the time the data block was written to the VBC 18 and the current time (on time input 68).

15 In Fig 3A the determined VBC 18 having the highest wait time is chosen by sequentially checking the wait times of each determined VBC 18 and storing only the single VBC having the highest (so far) wait time. This means that when all of the determined VBCs 18 have been checked, the
20 VBC 18 stored has the highest wait time and is therefore the VBC 18 from which data should be transmitted.

Steps 65w and 65x are used in the event that no VBCs 18 have data for transmission. In this case a VLF padded with zeros may be sent to give an indication of the
25 instantaneous channel bandwidth.

The selection output 80 then conveys information about the chosen VBC 18 to the integrator 64 (step 65y) which forms a VLF from the data block in the chosen VBC 18.

The number of data bits incorporated into the chosen
30 VBC 18 is then subtracted from the bit budget of the chosen VBC 18 (step 65z).

The procedure is then repeated beginning with step 65c and using the number of subtracted data bits in step 65z.

It will be appreciated that each VBC 18 is given an
35 initial bit budget. However, after a number of frames have been transmitted the bit budgets for each of the VBCs 18 may have changed considerably from the initial bit budget

18

values. This is because the bit budgets of the VBCs 18
which are part of the set of VBCs having data for
transmission are altered after every VLF is written, with
those VBCs 18 having the highest minimum bandwidth

5   receiving the greatest share of distributed bits after each
VLF is transmitted. The significance of this is a VBC 18
can be given a high priority by assigning it a high minimum
bandwidth.

It will also be apparent that if the initial

10 individual bit budgets (step 65b) are very small then this
will only allow small blocks to be written because the
total bit budget (which never changes) will be small;
whereas if the initial individual bit budgets are very
large then the wait time will greatly influence which data

15 block is transmitted because many of the VBCs 18 will have
sufficient bit budgets to transmit their data blocks. This
is why the initial values of individual bit budget for each
VBC 18 should not be very small or very large.

Fig 4 is a diagram which illustrates the

20 implementation of a VBC 18. Data can be written to the VBC
18 by a corresponding producer 16 in contiguous blocks of
one or more bytes, but if there is insufficient room to
perform a write, that is, if the VBC 18 is full, then the
write is blocked. A blocked write means that the write

25 cannot be performed until a certain condition arises or an
event happens, for example the VBC 18 has a certain amount
of free space or a read from the VBC 18 occurs. Data can
be read from the VBC 18 by the integrator 64. If there is
no data in the VBC 18 then the read will return a zero, it

30 will not block.

Fig 4 shows the VBC 18 as a circular buffer with a
write pointer 100, private data pointer (PDP) 101, and a
read pointer 102 peripherally spaced from the write pointer
100. The read pointer 102 is only repositioned after a

35 read from the VBC 18; and the write pointer 100 is only
repositioned after a write to the VBC 18. The VBC 18 is
empty when the read pointer 102 points to the same location

as the write pointer 100. The VBC 18 is full when the write pointer 100 points to the location immediately before the read pointer 102 in the clockwise direction.

The private data pointer (PDP) 101 is used because a producer 16 writes data to its associated VBC 18 one byte at a time. The multiplexer is prevented from reading out the data as it is being written, so the PDP 101 is used to indicate that the data that follows the PDP 101 is part of a data block which has not yet been completely written. Once the data block has been completely written the PDP 101 is updated to point to the location immediately after the data block that has just been written, and the characteristics of the newly written block are entered in table 19 (which is part of buffer 18). In table 19, the deposit time (column 19a) is the time that PDP 101 was updated, and the block size (column 19b) is measured by the angular movement of PDP 101. The entries in table 19 constitute the data identifier previously referred to.

Thus, the multiplexer 13 only has access to data which is located between the read index 102 and the PDP 101, and that data has an associated data identifier stored in table 19.

Each producer (for example 16a) is operable to interrogate the corresponding VBC 18a and thereby to determine the effective bandwidth of the VBC 18a, so that, in use, prior to writing a data block to the VBC 18a the producer 16a has determined the effective bandwidth and can control the rate at which data blocks are written to the VBC 18a and/or the size of data block to be presented to the VBC 18a. If there is a very low effective bandwidth in the VBC 18a then the producer 16a writes only small data blocks or it writes data blocks at less frequent intervals.

The bit-level implementation of data block multiplexing will now be described with reference to Figs 5 to 11.

Data blocks from different producers 16 are integrated by the multiplexer 13 into a sequence of VLFs. There are

20

two different types of VLFs: variable length control frames.
(VLCFs) and variable length data frames (VLDFs). VLCFs are
used by the multiplexer 13 to instruct the remote
de-multiplexer 14' on the number of distinct data types
5 which are encoded and what the data types are, for example
video, audio, midi, text, and such like. VLDFs encode a
variable number of data bytes from a single producer 16.

The format of a VLDF 104 is shown in Fig 5 and the
format of a VLCF 106 is shown in Fig 6. Each VLF has a
10 predetermined starting code (synchronising code) to
indicate that what follows is part of a VLF. Each VLF is
always byte aligned, that is, each VLF contains an integral
number of bytes of data and the predetermined starting code
always starts at the first bit of a byte.

15   In Fig 5 the predetermined starting code is a one byte
code (producer code) 110; whereas in Fig 6 the
predetermined starting code (control code) 112 is in the
form of two successive occurrences of the producer code
110. The producer code 110 is the binary value 10010011.
20 As this code 110 is reserved for indicating either a VLDF
or a VLCF any occurrences of this code 110 not intended to
indicate the start of a VLDF must be avoided. This is done
using a procedure called bit stuffing, which will be
described in more detail below.

25   The second field in the VLDF is a channel identity
field 114. The channel identity field 114 indicates the
identity of the producer 16 which generated the data block
incorporated in the VLDF. Up to seven bits are allocated
as the channel identity field 114.

30   If, for example, only four producers 16 are active
then only the two most significant bits are required to
identify uniquely each producer 16.

The third field is the data field 116 which contains
a data block from one of the VBCs 18. The data field 116
35 begins at the first free bit of the channel identity field
114. The data field 116 may not be the same length as the
original data block because bit stuffing is used to avoid

21

the producer code 110 occurring within the data field 116.
The meaningful data (indicated by numeral 118) in the data
field 116 may end at any bit position within the last byte;
therefore, zero bit padding 120 may be used in the last
5   byte to ensure that the entire VLDF is a whole number of
bytes.

The VLCF has a control code field 112 occupying the
first two bytes and a channel command field 130 immediately
after the control field 112. The channel command field 130
10   is typically three to six bytes in length, and in this
embodiment is four bytes. Examples of the codes used by
the channel command field 130 and the interpretation of the
codes are given in Figs 7a and 7b.

In Fig 7a, the add channel code 132 is shown. This
15   code 132 is used to add a producer 16 to the system, that
is to identify a new producer 16 as active and consequently
requiring a corresponding consumer 20'. Conversely, the
delete channel code 134 is used to identify a currently
active producer 16 as inactive so that the corresponding
20   currently active consumer 20' can be de-activated. The
restart code 136 is used to identify all of the producers
16 as de-activated; and the terminate code 138 is used to
identify that the transmission is to be stopped.

Immediately after the channel command field 130 there
25   is a parameters field 140 which is typically between zero
and five bytes in length depending on the parameters stored
therein. As shown in Fig 7b, when the add channel code 132
is used, the parameters add channel field 142 has a seven
bit identifier 146 to identify which local producer 16 is
30   being activated and a four byte data ASCII string
identifier 148 to indicate what type of data (for example,
"SCT*", "MIDI", "TEXT") will be transmitted from that
producer 16. This identifier is used by the remote
de-multiplexer 14' to provide each remote consumer 20' with
35   the appropriate data.

When the delete channel code 134 is used then the
parameters delete channel field 144 has a seven bit

22

identifier 146 to identify which local producer 16 (and consequently which corresponding remote consumer 20') is being de-activated. There are no parameters fields associated with the restart code 136 or the terminate code 138.

Byte stuffing is illustrated in Figs 8 to 11.

Fig 8a shows how the multiplexer 13 re-organises data if the producer code 110 accidentally occurs within the data field 116. If the producer code 110 does occur in a byte within the data field 116 then the multiplexer 13 substitutes a conversion code 150 for the producer code 110. The conversion code 150 is the binary value 10111011. As the VLDF is byte aligned any occurrence of the digits of the producer code 110 which straddle part of two bytes would not be recognised as a producer code 110.

A problem arises if the conversion code 150 occurs in a byte within the data field 116 because the multiplexer 13 would not know whether the conversion code 150 was the correct sequence or whether the producer code 110 was intended.

To overcome the potential ambiguity, whenever the first seven bits (the most significant seven bits) of the conversion code 150 occur at the start of a byte, other than when a producer code 110 has been converted to the conversion code 150, then the eighth bit is set to zero and the original eighth bit is transferred to the first bit of the next byte as shown in Fig 8b. The first seven bits of the conversion code are termed the escape code 152.

Fig 9 illustrates the construction of a VLDF 104. The first byte is the producer code 110. The first three bits of the second byte comprise the channel identity field 114 (for identifying up to eight producers 16). The data block to be integrated 160 is then shuffled and packed into the data field 116. As the first three bits of the second byte are used for the channel identity field 114, the next five bits are used to store the last five bits of the first byte of the data block to be integrated 160. The first three

bits of the data block 160 are then stored in the first three bits of the third byte. This process continues until the entire data block 160 has been written to the data field.

Fig 10 illustrates this process using specific binary values as an example. Wherever the escape code 152 occurs within the data field 116 then the change described with reference to Fig 8b occurs.

Fig 11 illustrates what happens if the data field 116 finishes part of the way through a byte: the remaining bits in that byte are set to zero (zero bit padding). The next byte will be a producer code 110 for the next VLDF. There is no ambiguity caused by the additional zero or zeros because the data field 116 is always a whole number of bytes so any additional bits (the added zeros) are ignored when the data field 116 is decoded.

Fig 12 illustrates the de-multiplexer 14. The de-multiplexer 14 receives a sequence of variable length frames via the buffer 34 (Fig 2) and then decodes this sequence of VLDFs 104 and/or VLCFs 106 by reversing the encoding process described with reference to Figs 5 to 11.

A read from buffer 34 may be blocking or non-blocking. A non-blocking read will return up to the number of bytes requested, or zero if there was no data in the buffer 34; whereas, a blocking read will wait until there are the requested number of bytes in the buffer 34 and will return those bytes.

On power-up, each local consumer 20 is registered with the de-multiplexer 14, and each local consumer 20 specifies the type of data that it can receive using a four character ASCII string identifier (for example, "SCT*", "MIDI", "TEXT") identical to identifier 148.

On registration, a consumer DSB 22 will be assigned to each consumer 20; and the consumer 20 will be placed on a list of local consumers 20 available to connect to a remote producer 16' generating data of the same type (as indicated by identifier 148). Several consumers 20 may receive data

24

of the same type, but there is always a one-to-one correspondence (a single producer to a single consumer link) between remote producers 16' and local consumers 20.

If a VLCF 106 is received by the de-multiplexer 14 then the channel command field 130 is read and the appropriate action is taken.

If the channel command field 130 has an add channel command then identifier 148 is read and compared with the type of data which can be received by the registered consumers 20 which are available to connect but are not yet connected. If a match is found then the consumer DSB (for example 22m) assigned to the matched consumer 20m is connected to the de-multiplexer 14 and linked to the remote producer 16. The matched consumer 20m is removed from the list of local consumers 20 available to connect to a remote producer 16'.

If the channel command field 130 has a delete channel command then the de-multiplexer 14 disconnects the consumer DSB 22 associated with the local consumer 20 corresponding to the remote producer 16' identified by the seven bit identifier 146; this has the effect of breaking the link between the remote producer 16' and the local consumer 20. The local consumer 20 can still read all data currently in its associated DSB 22, however, no new data can be written to that DSB 22 until a request is received to re-assign that consumer 20 to the available to connect list, that is, until the consumer 20 is linked to a remote producer 16.

If the channel command field 130 has a restart command, then all connections between the de-multiplexer 14 and the consumer DSBs 22 (and thereby the links between the remote producers 16' and the local consumers 20) are removed, and the consumers 20 are returned to the available to connect list. This command is used to recover the system 12 after a severe burst error. Following a restart, the multiplexer 13 should re-establish the connections using the add channel command. The consumer DSBs 22 are not set to indicate that the sequence of VLFs has

25

terminated.

If the channel command field 130 has a terminate command, then all connections between the de-multiplexer 14 and the consumer DSBs 22 are removed and the consumer DSBs 22 are set to indicate that the sequence of VLFs has terminated. Each local consumer 20 can still read all data currently in its associated DSB 22, however, no new data can be written to the consumer DSBs 22 until a request is received to re-assign the consumers 20 to the available to connect list.

If a VLDF 104 is received by the de-multiplexer 14 then the channel identity field 114 is read and the corresponding (linked) local consumer 20 is assigned the data that follows in the data field 116, that is, the data in the data field 116 is written to the associated consumer DSB 22. If there is no local consumer 20 identified by the channel identity field 114 then the data that follows is discarded.

If the write to the consumer DSB 22 fails then the local consumer 20 may request that the link be terminated, in which case the consumer DSB 22 is disconnected from the de-multiplexer 14. The de-multiplexer 14 also sends a request to the remote multiplexer 13' to cease transmission from the corresponding remote producer 16'.

If the write to one of the consumer DSBs 22 fails because the DSB 22 is full, then the data is discarded and a count is kept of the bytes that are lost until the number of lost bytes exceeds a predetermined limit. Once this occurs, the de-multiplexer 14 disconnects the consumer DSB 22, thereby removing the link between the local consumer 20 and the remote producer 16'. The de-multiplexer 14 also sends a request to the remote multiplexer 13' to cease transmission from that producer 16'.

It is the responsibility of the producer 16 and/or the consumer 20' to determine whether any errors have occurred in transmitting the data. It is not the responsibility of the multiplexer 13 or de-multiplexer 14 to locate or

26

correct any transmission errors.

·Fig 13 is a schematic diagram of an alternative data transmission system 200 which includes a supervisor 202 which connects the de-multiplexer 14 to the multiplexer 13 via VBC 42. System 200 thereby enables a local consumer 20 to communicate with a remote producer 16'.

System 200 allows a more flexible prioritisation strategy to be implemented than is possible with system 12. For example, system 200 enables the buffer identifiers to be modified in use, thereby enabling the prioritisation of the buffers and the bandwidth assignment of the channel arbiter 60 to be altered.

The supervisor 202 performs a number of tasks to control the operation of the data transmission system 200. The supervisor 202 creates and configures the producers 16 and consumers 20 when the system 200 is powered-up. The supervisor 202 determines how many producers 16 are active and instructs the multiplexer 13 as to how many bits are required for the channel identity field 114. The supervisor 202 routes messages between the local/remote multiplexer/de-multiplexer. The supervisor 202 is also responsible for terminating all processes within the data transmission system 200.

Various modifications may be made to the above described embodiments within the scope of the present invention, for example, any convenient channel may be used rather than a GSM telephone line. The channel may be a physical channel such as a cable. In other embodiments data transmission system 12 may have a different number of producers 16 than data transmission system 12'; similarly, data transmission system 12 may have a different number of consumers 20 than data transmission system 12'; for example, in remote system 12', there are n consumers 20' and m producers 16'; whereas, in local system 12 there may be n producers 16 and m consumers 20. In other embodiments, different predetermined strategies may be used for determining the order in which data blocks should be

formed into VLFs for transmission.

28

CLAIMS

1.   A multiplexer for transmitting real-time data from a
plurality of producers to a plurality of corresponding
consumers over a channel, where each producer is operable
to present a data block for transmission, so that at any
point in time a plurality of data blocks are presented for
transmission,

      the multiplexer being arranged in use to create and
transmit across the channel a data stream comprising a
sequence of variable length frames, where each frame
contains either all or part of one of the plurality of data
blocks and does not contain data from any other data block.

2.   A multiplexer as claimed in claim 1, wherein a
variable length frame (VLF) comprises a predetermined code,
an identifier for indicating which consumer the data is
intended for, and the data to be transmitted.

3.   A multiplexer as claimed in either preceding claim,
wherein each variable length frame is byte aligned so that
the predetermined code begins at the start (the first bit)
of a byte.

4.   A multiplexer as claimed in any preceding claim,
wherein   the   multiplexer   is   incorporated   in   a
multiplexer/de-multiplexer data transmission system for
duplex transmission and reception of data.

5.   A multiplexer/de-multiplexer data transmission system
as claimed in claim 4, wherein the data transmission system
further comprises buffer means associated with each of the
plurality of producers, where the buffer means stores data
for transmission.

6.   A system as claimed in claim 5, wherein at least one
producer is an adaptive producer which is operable to
regulate either the time interval at which it presents data

29

blocks to the buffer means for transmission, and/or the size of data block which it presents to the buffer means for transmission.

7.   A system as claimed in claim 6, wherein the adaptive producer is operable to interact with the buffer means and thereby to determine the effective bandwidth of the buffer means (that is, the rate at which data is transferred from the buffer means to the multiplexer), whereby, in use, prior to writing a data block to the buffer means the adaptive producer interrogates the buffer means to determine the effective bandwidth of the buffer means, and the adaptive producer accordingly controls the size of data block to be written to the buffer means and/or the rate at which data blocks are written.

8.   A system as claimed in claim 7, wherein each data block written to the buffer means has a data identifier.

9.   A system as claimed in claim 8 wherein the buffer means comprises: a buffer for holding data blocks which are to be transmitted, and a memory for holding the data identifiers associated with the data blocks in that buffer; the buffer means having a FIFO (first in first out) arrangement so that each buffer means presents only one data block and the associated identifier at a time (the current block and identifier) to the multiplexer.

10. A system as claimed in claim 9, wherein the multiplexer is adaptive and communicates with the transmission channel via a variable bandwidth FIFO buffer and is operable to regulate the time interval at which it presents VLFs to the buffer and/or the size of VLFs according to the effective bandwidth of the variable bandwidth FIFO buffer.

11. A method of transmitting real-time data from a

30

plurality of producers over a channel, the method comprising the steps of:

conveying from each of the producers a data block to be transmitted from one of the producers;

creating a variable length frame using one conveyed data block; and

transmitting the variable length data frame across the channel.

12. A method as claimed in claim 11, wherein the creating step further comprises the step of interrogating the identifier associated with each presented data block to determine the data block which has highest priority according to a predetermined strategy and selecting this highest priority data block for conveying.

13. A method as claimed in claim 12, wherein the predetermined strategy is a function of channel bandwidth (which may vary from time to time), buffer identifier (which may be preset or vary from time to time), and data block identifier (which indicates the block length which may vary from buffer to buffer and also may vary from time to time in any one buffer).
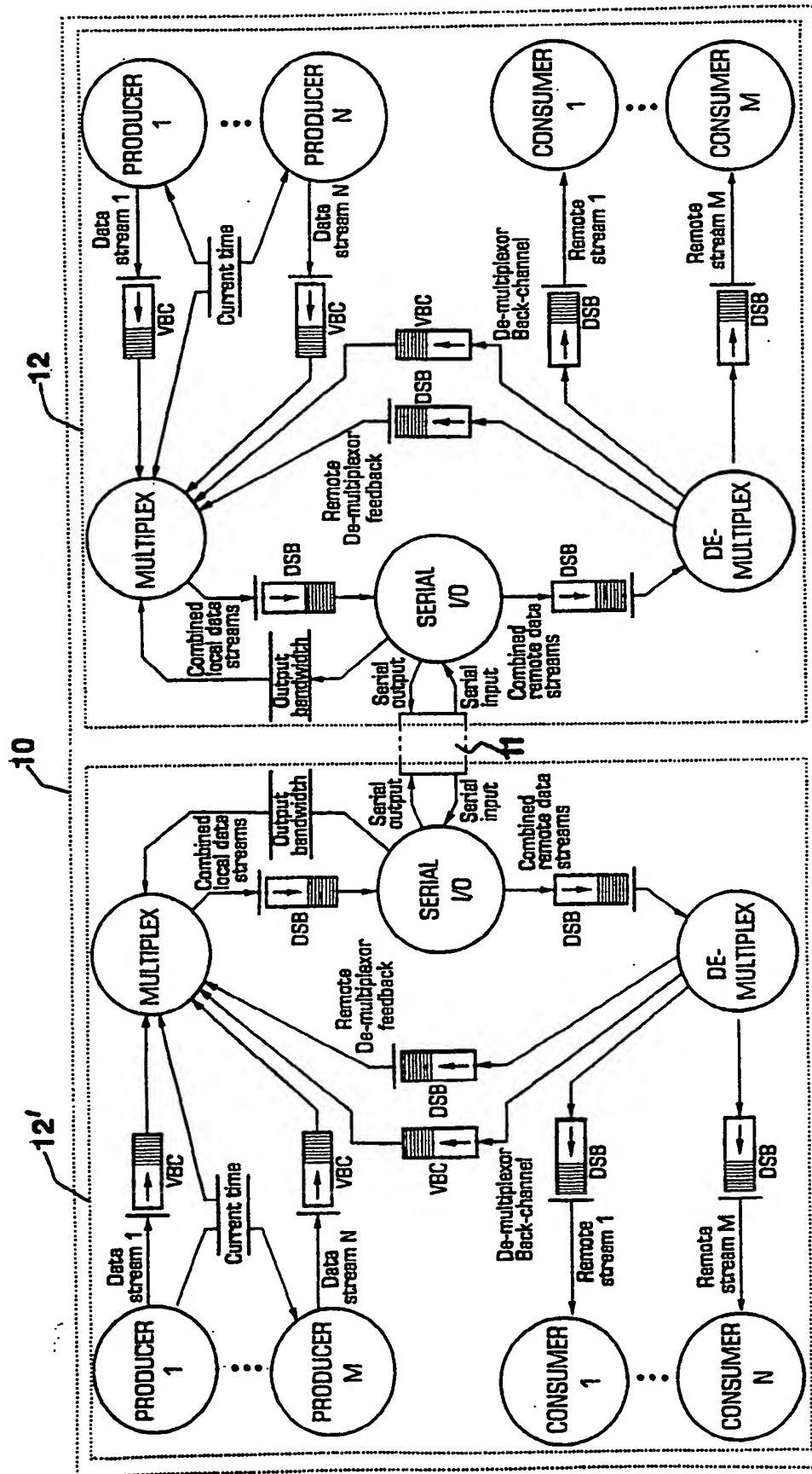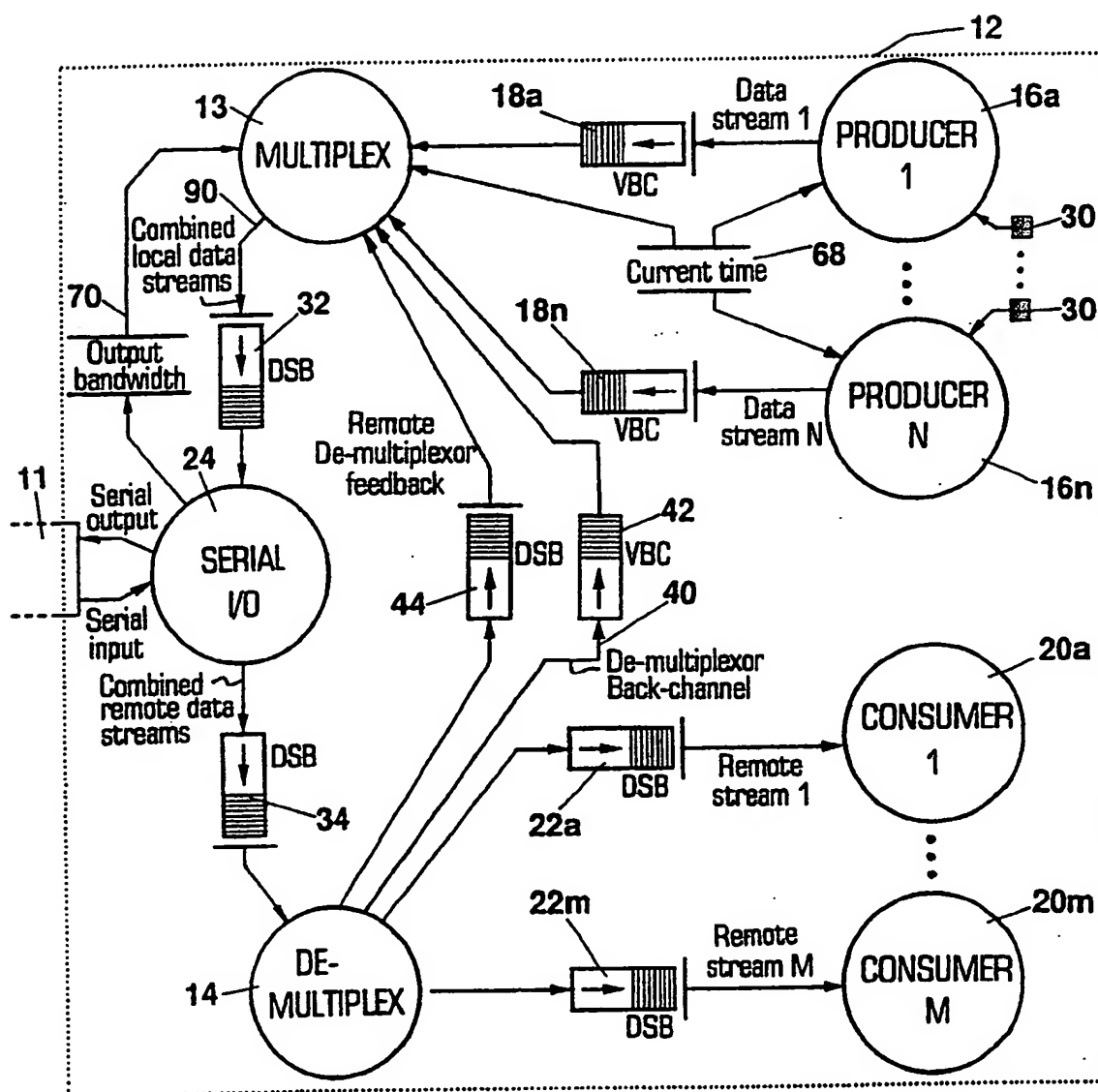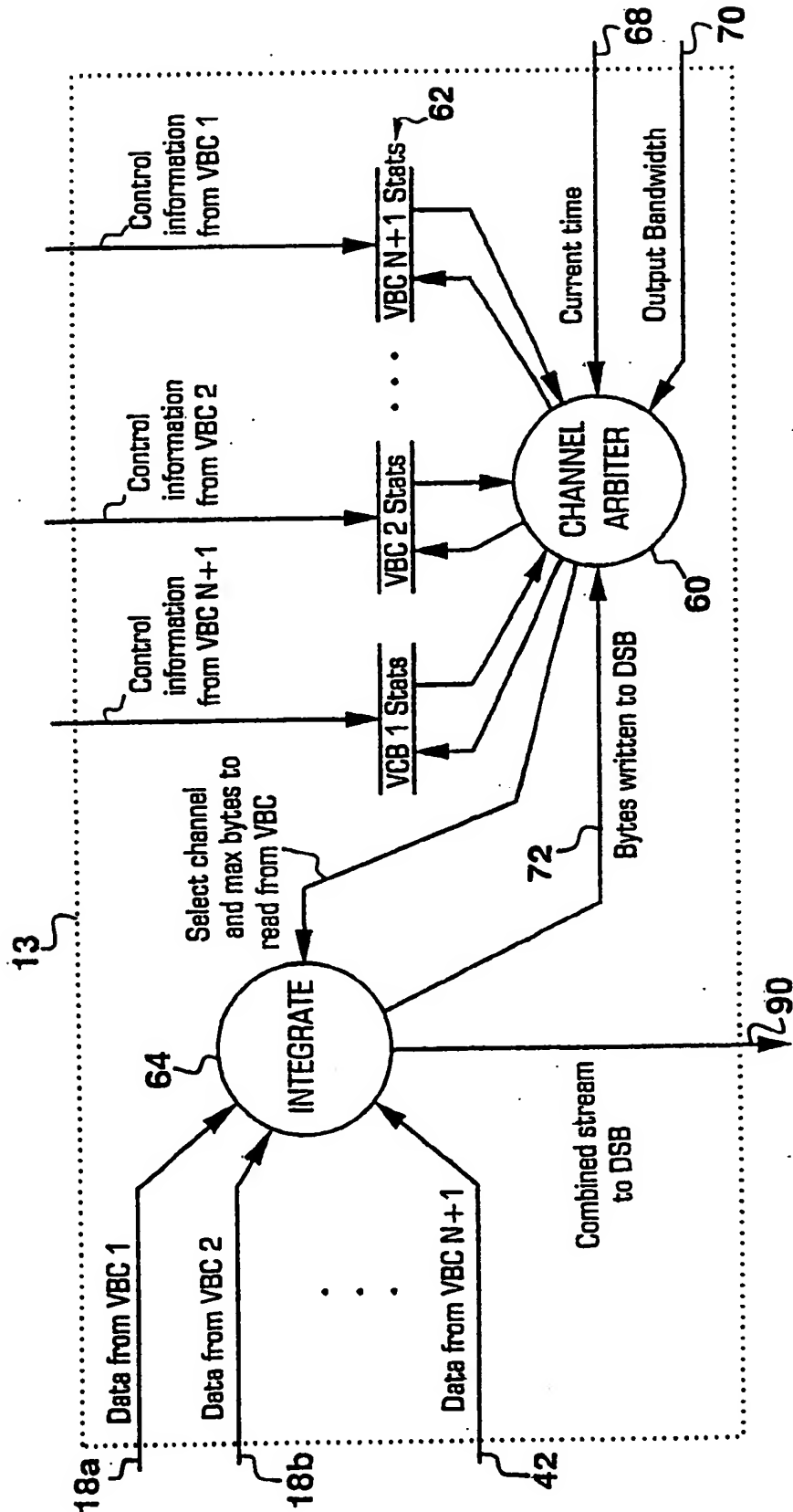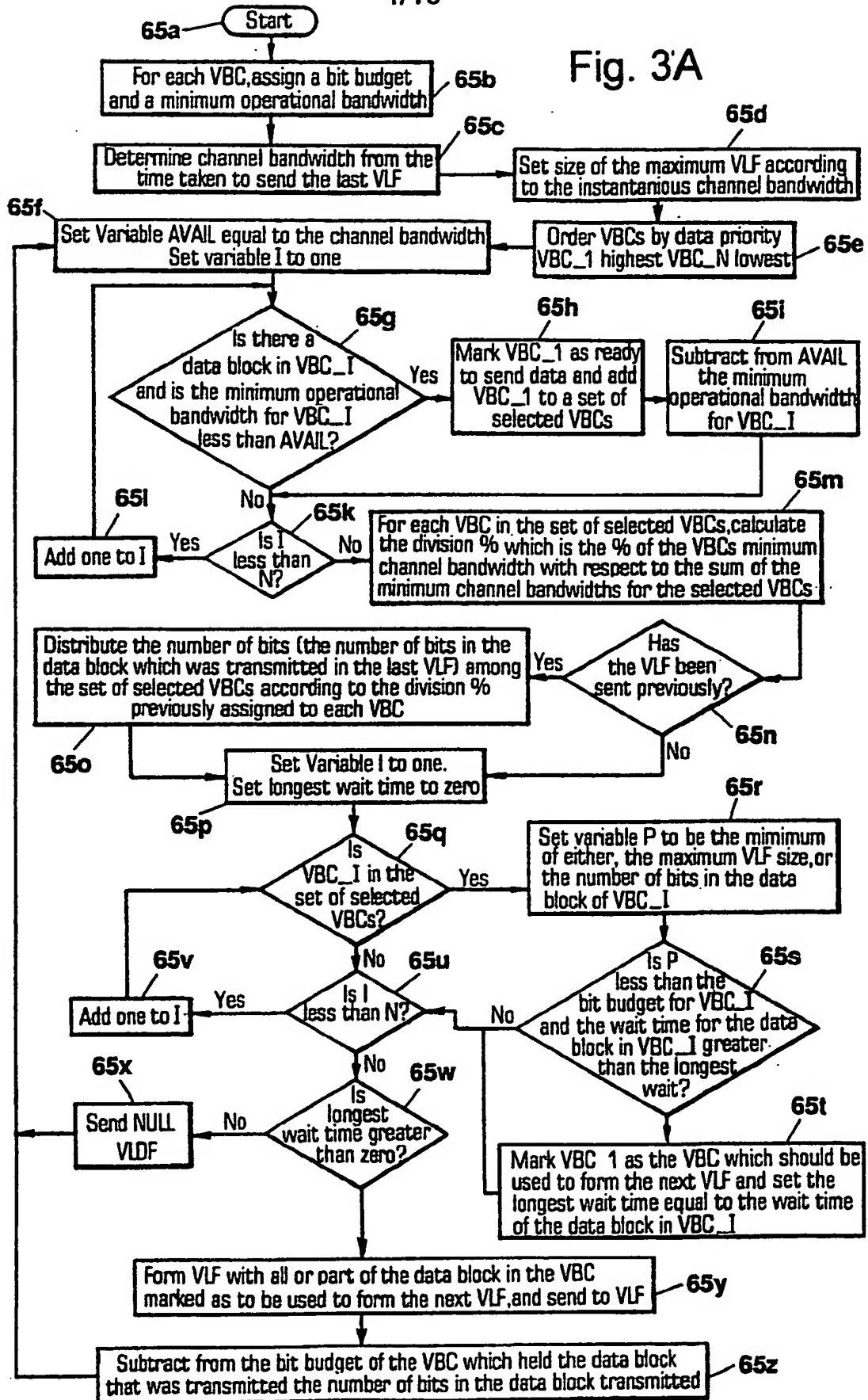
Fig. 1

Fig. 2

Fig. 3

4/10

Fig. 3'A

65a ○ Start

65b For each VBC, assign a bit budget and a minimum operational bandwidth

65c Determine channel bandwidth from the time taken to send the last VLF

65d Set size of the maximum VLF according to the instantanious channel bandwidth

65f Set Variable AVAIL equal to the channel bandwidth Set variable I to one

65e Order VBCs by data priority VBC_1 highest VBC_N lowest

65g Is there a data block in VBC_I and is the minimum operational bandwidth for VBC_I less than AVAIL?

Yes →

65h Mark VBC_1 as ready to send data and add VBC_1 to a set of selected VBCs

65i Subtract from AVAIL the minimum operational bandwidth for VBC_I

No ↓

65l Add one to I ← Yes

65k Is I less than N? No →

65m For each VBC in the set of selected VBCs, calculate the division % which is the % of the VBCs minimum channel bandwidth with respect to the sum of the minimum channel bandwidths for the selected VBCs

65o Distribute the number of bits (the number of bits in the data block which was transmitted in the last VLF) among the set of selected VBCs according to the division % previously assigned to each VBC

Yes ← Has the VLF been sent previously?

65n No ↓

65p Set Variable I to one. Set longest wait time to zero

65q Is VBC_I in the set of selected VBCs? Yes →

65r Set variable P to be the mimimum of either, the maximum VLF size, or the number of bits in the data block of VBC_I

No ↓

65v Add one to I ← Yes

65u Is I less than N? No →

65s Is P less than the bit budget for VBC_I and the wait time for the data block in VBC_I greater than the longest wait?

No ↓

65w Is longest wait time greater than zero?

65x Send NULL VLDF ← No

65t Mark VBC 1 as the VBC which should be used to form the next VLF and set the longest wait time equal to the wait time of the data block in VBC_I

65y Form VLF with all or part of the data block in the VBC marked as to be used to form the next VLF, and send to VLF

65z Subtract from the bit budget of the VBC which held the data block that was transmitted the number of bits in the data block transmitted

Fig. 4

Fig. 5

Fig. 6

Command	32-bit code	Command Description	
ADD CHANNEL	14141414_{16}	Add a new virtual channel from a producer to the integration of producer data	132, 134
DELETE CHANNEL	22222222_{16}	Remove the virtual channel for a producer from the integration of producer data	136
RESTART	48484848_{16}	Mark all virtual channels as removed from the integrated stream	138
TERMINATE	81818181_{16}	Mark end of integrated stream	

Fig. 7a

Following the 32-bit command are the command parameters as follows:

Command	Parameters 146 — 148	
ADD CHANNEL	<7-bit channel id> <4-byte data identifier>	142
DELETE CHANNEL	<7-bit channel id>	144
RESTART	NONE 146	
TERMINATE	NONE	

Fig. 7b
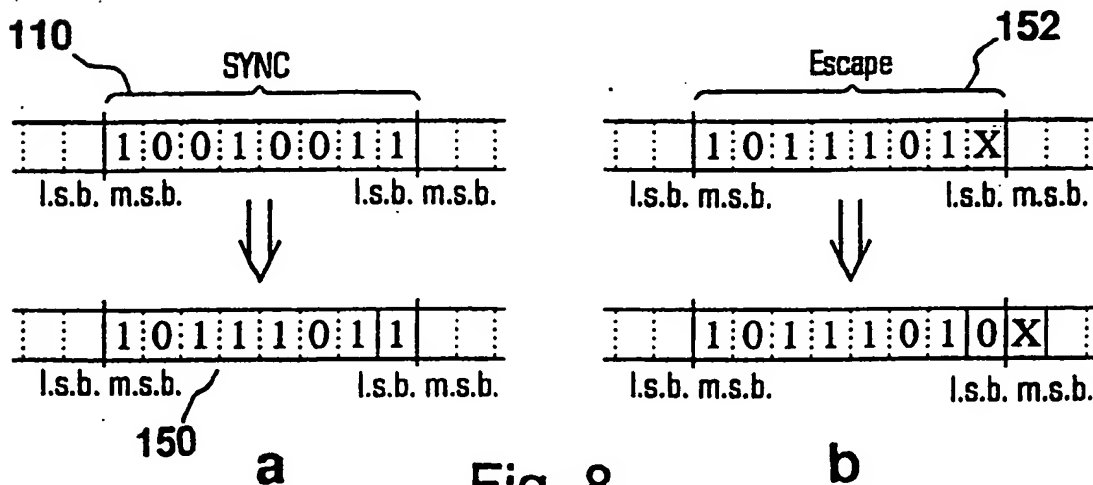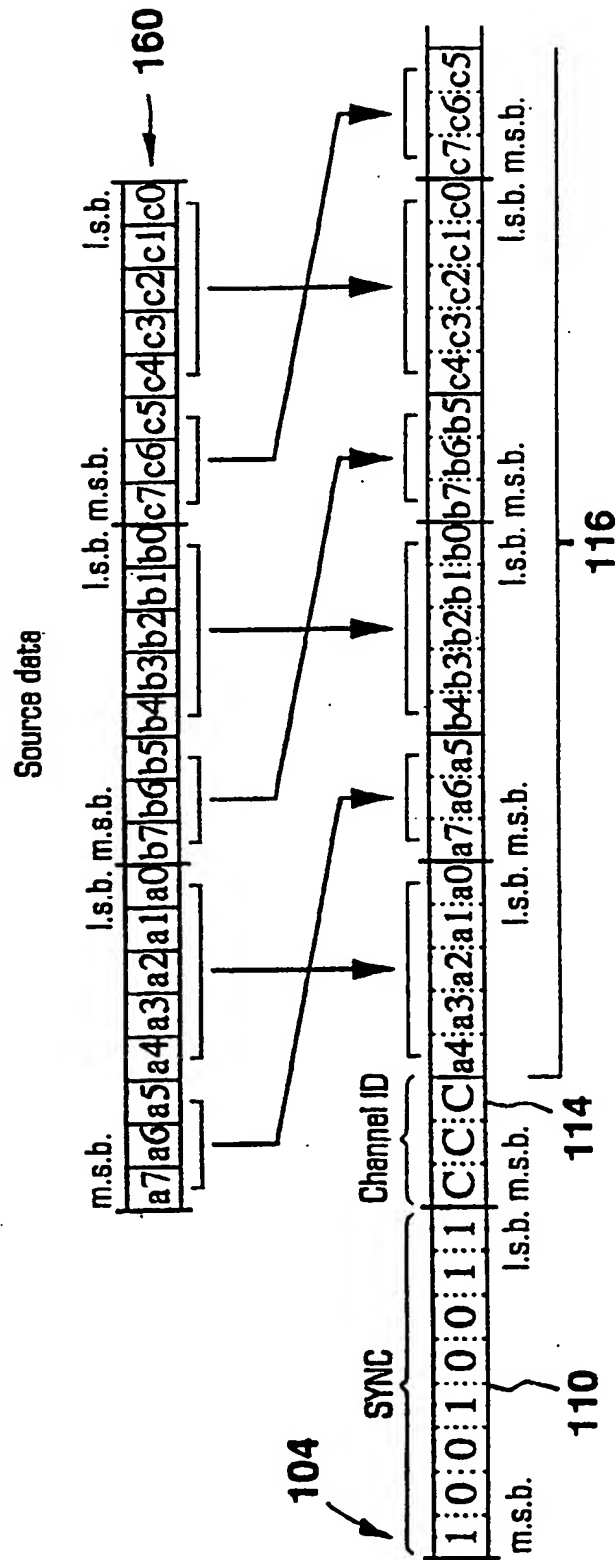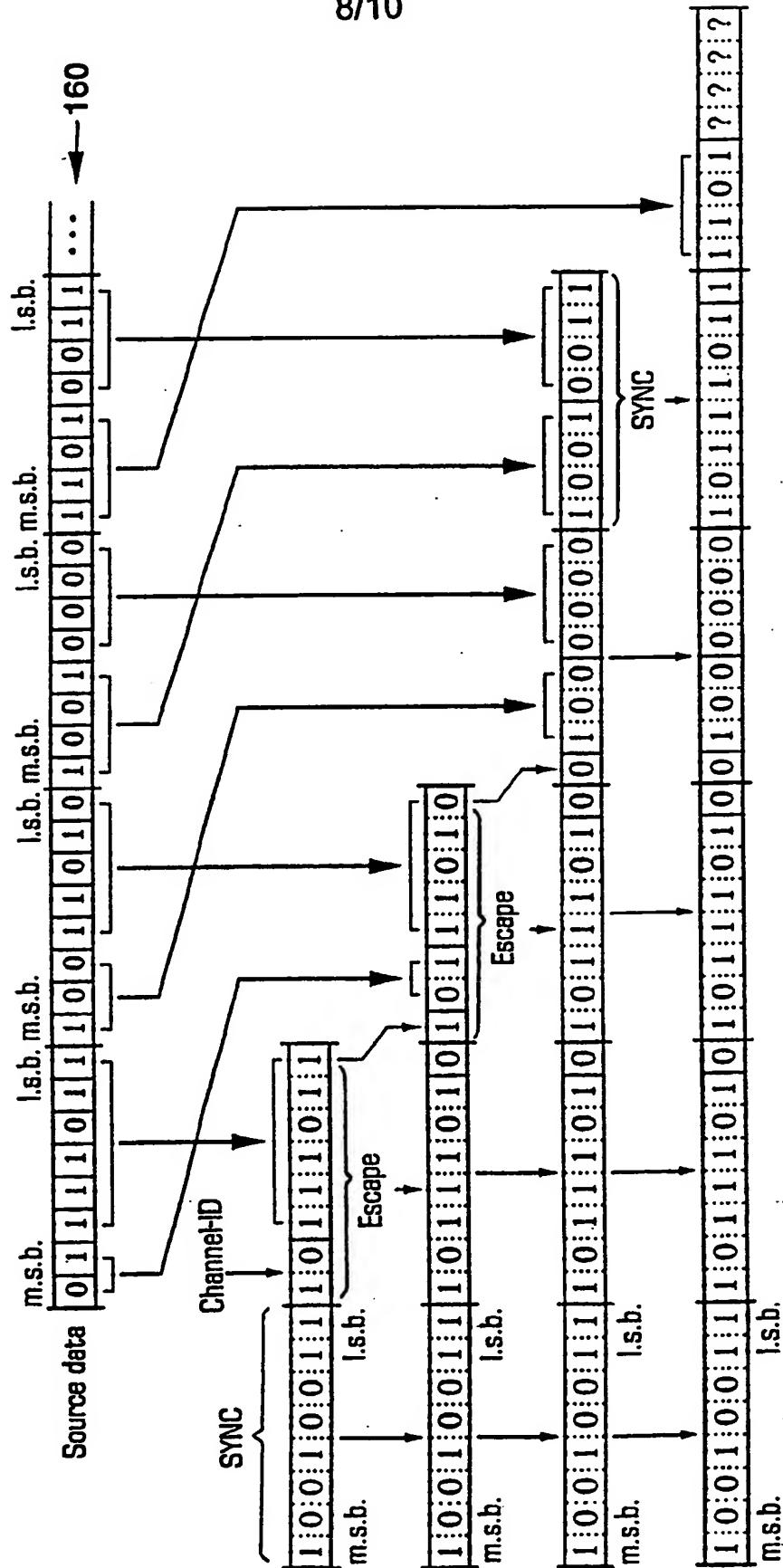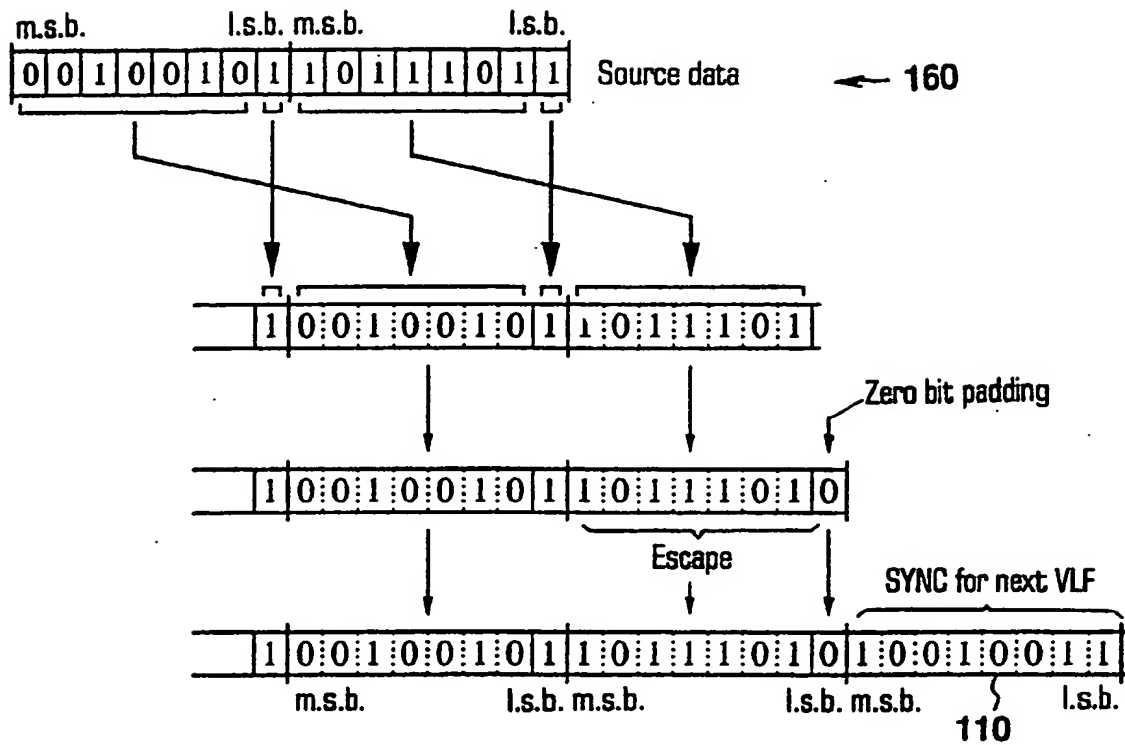
110   SYNC   152   Escape

1 0 0 1 0 0 1 1   1 0 1 1 1 0 1 X

l.s.b. m.s.b.   l.s.b. m.s.b.   l.s.b. m.s.b.   l.s.b. m.s.b.

⇓   ⇓

1 0 1 1 1 0 1 1   1 0 1 1 1 0 1 0 X

l.s.b. m.s.b.   l.s.b. m.s.b.   l.s.b. m.s.b.   l.s.b. m.s.b.

150

a   **Fig. 8**   b

Fig. 9

Fig. 10

Fig. 11
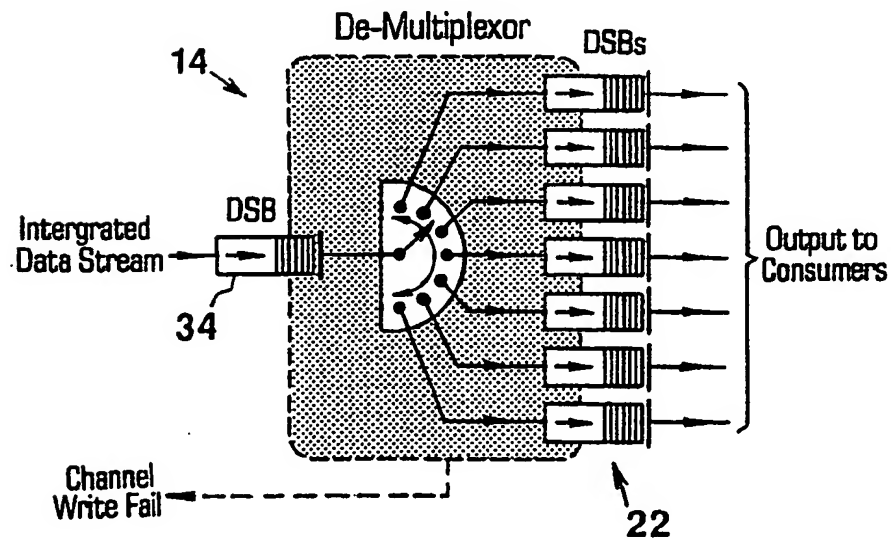
Fig. 12

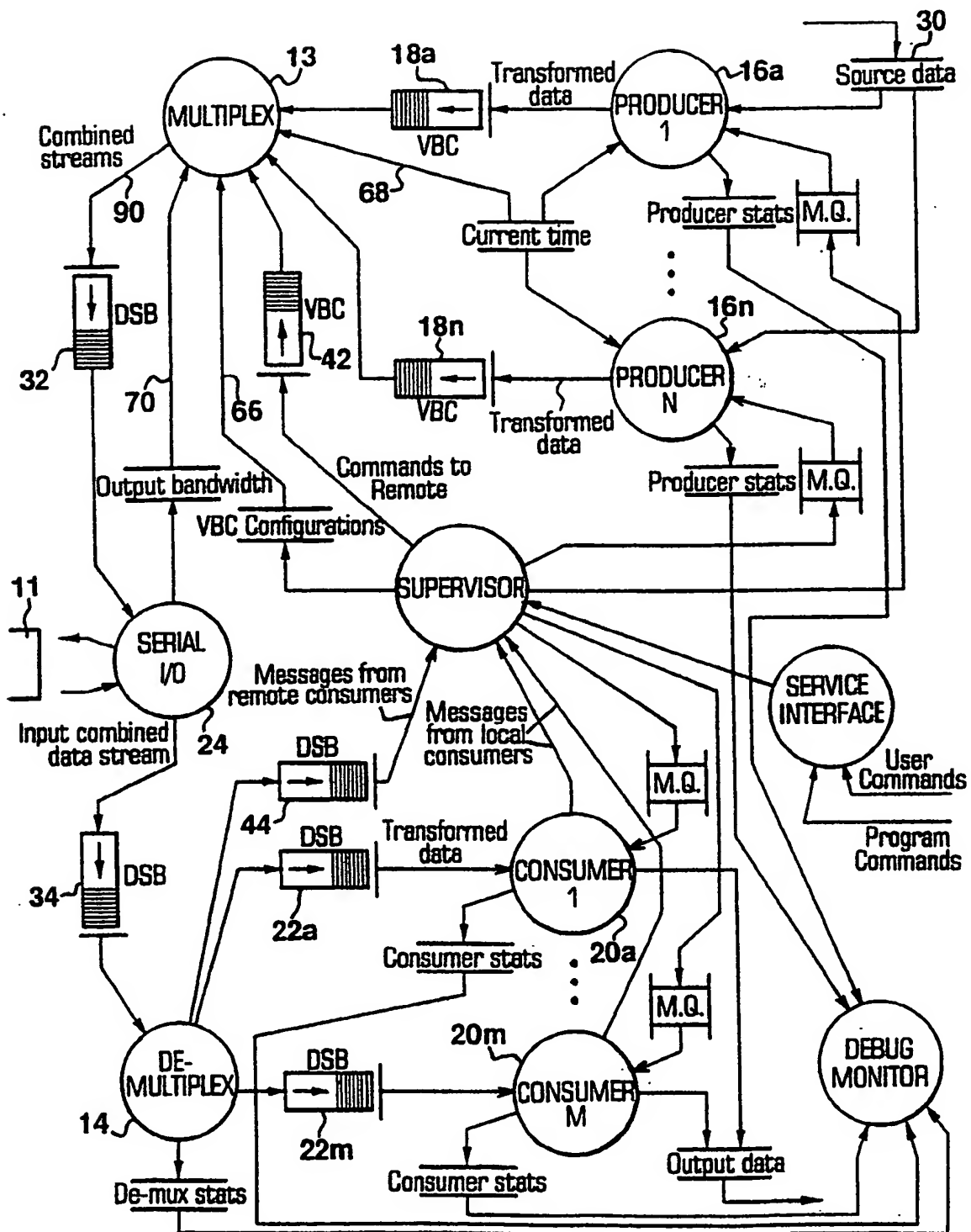Fig. 13

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC 6 H04J3/24   H04L29/06   H04N7/58

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6   H04J

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category°	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 541 919 A (YONG MEI ET AL) 30 July 1996 (1996-07-30) abstract column 1, line 12 - column 2, line 10 column 2, line 53 - column 3, line 26 column 3, line 53 - column 4, line 15 column 9, line 14 - line 42	1-13

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 July 1999	06/08/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Chauvet, C

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 5541919	A	30-07-1996	CA	2182296 A	27-06-1996
			CN	1141106 A	22-01-1997
			EP	0745295 A	04-12-1996
			WO	9619882 A	27-06-1996